UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER OF PATENTS AND TRADEMARKS
Washington, D.C. 20231
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/329,558	06/10/1999	GRAHAM CHAPMAN	12463(CA998-	8251

7590 02/24/2003

RICHARD L CATANIA ESQ
SCULLY SCOTT MURPHY AND PRESSER
400 GARDEN CITY PLAZA
GARDEN CITY, NY 11530

EXAMINER	
TANG, KENNETH	

ART UNIT	PAPER NUMBER
2127	

DATE MAILED: 02/24/2003

Please find below and/or attached an Office communication concerning this application or proceeding.

PPG

Office Action Summary	Application No.	Applicant(s)	
	09/329,558	CHAPMAN ET AL.	
	Examiner	Art Unit	
	Kenneth Tang	2127	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133).
- Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *13 December 2002* .

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-23* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-23* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *10 June 1999* is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

11)☐ The proposed drawing correction filed on _____ is: a)☐ approved b)☐ disapproved by the Examiner.

If approved, corrected drawings are required in reply to this Office action.

12)☐ The oath or declaration is objected to by the Examiner.

Priority under 35 U.S.C. §§ 119 and 120

13)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____ .

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

14)☐ Acknowledgment is made of a claim for domestic priority under 35 U.S.C. § 119(e) (to a provisional application).

a) ☐ The translation of the foreign language provisional application has been received.

15)☐ Acknowledgment is made of a claim for domestic priority under 35 U.S.C. §§ 120 and/or 121.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☒ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449) Paper No(s) _____ .
4) ☐ Interview Summary (PTO-413) Paper No(s). _____ .
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: .

DETAILED ACTION

1. This final action is in response to paper number 4, Ammendment A, which was received on December 13, 2002. Applicant's arguments have been fully considered but they are not deemed to be persuasive. Claims 1-23 are presented for examination.

Claim Rejections - 35 USC § 102

(e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

2. Claims 1-3, 10-13, and 20-23 are rejected under 35 U.S.C. 102(e) as being unpatentable over Agesen (US 6,047,125).

Referring to claims 1 and 11, Agesen teaches the following limitations:

- simulating stack actions for executing bytecodes along said path ("bytecode", "stack", "execution", col 3, lines 62-65, and "pushed on stack by bytecode", "execution", col 11, lines 59-63)
- mapping a path of control flow on the stack from any start point in a selected method to the destination program counter by locating a linear path from the beginning of the method to the destination program counter and iteratively processing a bytecode sequence for each branch until said destination program counter is reached.

Agesen discloses using a "stack map" to map controls to a "program counter" (col 7, lines 1-5). There is an "active frame pointer" that is located on the stack and is used in part of the mapping. The pointer can be addressed at any of the locations (col 8, lines 53-60). It is inherent that the counter counts with integer iterations just like every other program counter can. For example, iterator "i" will keep iterating in a loop until the "if statement" condition is satisfied and if not, the iterator can be increased or decreased by 1. Specifically, if i is initially set to 0, and the "if statement" condition is "i=4," the iterator will complete the loop without satisfying the condition, update the iterator to "i=1," and keep looping until the iterator "i" completes the loop at "i=4."

Referring to claims 2 –3, and 12-13, Agesen teaches:

- processing a first linear bytecode sequence and an additional one until the control flow is interrupted ("two or more bytecode sequences", col 5, lines 19-27, and "Interrupt controller", Fig. 3A 135, col 7, lines 42-44)
- recording unprocessed targets from any branches in the first and additional linear bytecode sequence for future processing (Exception handler 760, bytecode 732, "stores a reference in variable 1", "branch to bytecode", col 11, lines 54-59).

Ageson inherently teaches:

- the destination program counter was not reached during an earlier processing of a linear bytecode sequence
- the information stored in the reference can be used at a future time for processing.

Ageson does teach that the program counter was reached during the later processing of a bytecode sequence (The bytecode 740 is stored in variable 2 value of the program counter, col 11, lines 45-47). In Ageson's reference, the program counter does not get reached in the earlier processing.

Referring to claims 10 and 20, Agesen teaches inserting pre-determined stack actions for bytecodes maintaining the control flow ("Bytecode 702 is used to push the integer value 1 on top of the operand stack", col 11, lines 5-6) and calculating stack actions for bytecodes transferring the control flow (bytecodes, stack, i=i+itmp, col 11, lines 19-23).

Referring to claims 21 and 22, it is inherent that computer-readable memory can be used to store instructions.

Referring to claims 23, it is rejected for the same reasons as stated in the rejections of claims 1-3.

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

3. Claims 4-5, 14-15 are rejected under 35 U.S.C. 103(a) as being unpatentable over Agesen (US 6,047,125) in view of Agesen (US 5,909,579).

Referring to claims 4-5 and 14-15, Agesen (US 6,047,125) fails to explicitly teach:

- determining if a bytecode in any linear bytecode sequence is a breakpoint with a pointer to bytecode data;

However Ageson (US 5,909,579) teaches a "bytecode analyzer mechanism" which determines the changes which the bytecode makes to the live pointer locations (col 8, lines 20-24). In addition, the system has breakpoints or computation stops at bytecode boundaries for determining live pointer information (col 3, lines 10-13). It would have been obvious to one of ordinary skill in the art at the time the invention was made to include the determination of a breakpoint feature to the existing system of Ageson for the reason of improving control by having an interrupt mechanism.

Agesen (US 6,047,125) also fails to explicitly teach:

- replacing the breakpoint with the bytecode data

However Ageson (US 5,909,579) also teaches an encoded bytecode change to the breakpoint or program stack frame pointer (bytecode encoded, change, program stack frame live pointer, col 8, lines 25-30). It would have been obvious to one of ordinary skill in the art at the time the invention was made to include the feature of replacing the breakpoint with the bytecode data for the reason of giving the system more flexibility for optimizing garbage collection.

4. Claims 6-7, 9, 16-17, and 19 are rejected under 35 U.S.C. 103(a) as being unpatentable over Agesen (US 6,047,125) in view of Gosling (US 5,668,999).

Referring to claims 6 and 16, Agesen fails to explicitly teach generating a virtual stack from executing bytecodes along the path. However, Gosling discloses generating a "virtual stack" 344 from the step of simulating stack actions executing "bytecodes" (col 5, lines 21-29). It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the virtual stack feature to the existing system of Agesen for the reason of being able to have an imaginary stack to store bytecodes.

Referring to claims 7 and 17, Gosling discloses:

- storing the bitstring at a selected destination for use in memory management operations (virtual stack 344 stored by bytecode, col 5, lines 33-40).

Gosling does not disclose:

- encoding the virtual stack as a bitstring

However, it is common knowledge that bitstrings can be encoded from binary.

Referring to claims 9 and 19, Agesen teaches storing the bitstring to a pre-allocated area on the stack ("bytecode 704 stores the value (1) from the top of the operand stack", col 11, lines 5-12).

5. Claims 8 and 18 are rejected under 35 U.S.C. 103(a) as being obvious over Agesen (US 6,047,125) in view of Gosling (US 5,668,999) and further in view of O'Connor (US 6,098,089).

Referring to claims 8 and 18, Agesen in view of Gosling fails to explicitly teach storing the bitstring on a heap. However, from the reference of O'Connor, it is well-known in the state of the art that "garbage collection" of "heap-allocated storage" is an "attractive model for dynamic memory management" (col 1, lines 39-42).

ARGUMENTS

6. Applicant argues the rejections of claims 1-3 and 11-13 (found on page 7) that Ageson's invention involves rewriting code and that the present invention does not involve any rewriting of code at all. In response to applicant's argument that the references fail to show certain features of applicant's invention, it is noted that the features upon which applicant relies (i.e., no rewriting of code) are not recited in the rejected claim(s). Although the claims are interpreted in light of the specification, limitations from the specification are not read into the claims. See *In re Van Geuns*, 988 F.2d 1181, 26 USPQ2d 1057 (Fed. Cir. 1993).

7. Applicant argues on the first paragraph of page 7 that the present invention deals with the dynamic nature of the stack by leaving open a small area of memory in the stack and tagging it

for dynamic mapping and Ageson '125 does not do that at all. In response to applicant's arguments against the references individually, one cannot show nonobviousness by attacking references individually where the rejections are based on combinations of references. See *In re Keller*, 642 F.2d 413, 208 USPQ 871 (CCPA 1981); *In re Merck & Co.*, 800 F.2d 1091, 231 USPQ 375 (Fed. Cir. 1986). Examiner already admitted that Ageson '125 fails to explicitly teach that limitation. However, Ageson '125 in view of Gossling and futher in view of O'Connor.

8. Applicant argues on the third paragraph of page 7 that Agesen '125 teaches rewriting the Java code, not walking the stack. In response, Examiner respectfully disagrees. Agesen discloses using a "stack map" to map controls to a "program counter" (col 7, lines 1-5). There is an "active frame pointer" that is located on the stack and is used in part of the mapping. The pointer can be addressed at any of the locations (col 8, lines 53-60). It is inherent that the counter counts with integer iterations just like every other program counter can. For example, iterator "i" will keep iterating in a loop until the "if statement" condition is satisfied and if not, the iterator can be increased or decreased by 1. Specifically, if i is initially set to 0, and the "if statement" condition is "i=4," the iterator will complete the loop without satisfying the condition, update the iterator to "i=1," and keep looping until the iterator "i" completes the loop at "i=4."

9. Applicant argues on the first paragraph of page 8 that the Examiner has not shown an anticipation and does not appear to be supportable. In response to applicant's argument that there is no suggestion to combine the references, the examiner recognizes that obviousness can only be

established by combining or modifying the teachings of the prior art to produce the claimed invention where there is some teaching, suggestion, or motivation to do so found either in the references themselves or in the knowledge generally available to one of ordinary skill in the art. See *In re Fine*, 837 F.2d 1071, 5 USPQ2d 1596 (Fed. Cir. 1988)and *In re Jones*, 958 F.2d 347, 21 USPQ2d 1941 (Fed. Cir. 1992). In this case, however, Applicant also arguing limitations not stated in some of the claims.

10. Applicant argues on paragraph 2 of page 8 that Ageson '579 is directed to a system that chooses to store full information for only every nth bytecode in the stream, and recreating the deltas in between them (from Abstract). In response, Ageson '579 directs more than that. Ageson '579 also has a "live pointer information for the nearest bytecode preceding the desired bytecode boundary is retrieved along with the intervening coded changes" (see Abstract). In addition, However Ageson (US 5,909,579) teaches a "bytecode analyzer mechanism" which determines the changes which the bytecode makes to the live pointer locations (col 8, lines 20-24). In addition, the system has breakpoints or computation stops at bytecode boundaries for determining live pointer information (col 3, lines 10-13). It would have been obvious to one of ordinary skill in the art at the time the invention was made to include the determination of a breakpoint feature to the existing system of Ageson for the reason of improving control by having an interrupt mechanism. Furthermore, Ageson (US 5,909,579) also teaches an encoded bytecode change to the breakpoint or program stack frame pointer (bytecode encoded, change, program stack frame live pointer, col 8, lines 25-30). It would have been obvious to one of ordinary skill in the art at the time the invention was made to include the feature of replacing the

breakpoint with the bytecode data for the reason of giving the system more flexibility for optimizing garbage collection.

11. Applicant argues on the third paragraph of page 8 that the virtual stack thus created is only a stack for part of the program, unlike Gosling '999 which creates a virtual stack for the entire program "used in the same way as a regular stack." In response to applicant's argument that the references fail to show certain features of applicant's invention, it is noted that the features upon which applicant relies (i.e., "the virtual stack thus created is only a stack for part of the program") as recited in the rejected claim(s). Although the claims are interpreted in light of the specification, limitations from the specification are not read into the claims. See *In re Van Geuns*, 988 F.2d 1181, 26 USPQ2d 1057 (Fed. Cir. 1993).

12. Applicant argues on the first paragraph of page 9 that although storing information to a pre-allocated area on the stack is known in general, the claimed pre-allocated area is used when a special event occurs. In response to applicant's argument that the references fail to show certain features of applicant's invention, it is noted that the features upon which applicant relies (i.e., "pre-allocated area is used when a special event occurs") as recited in the rejected claim(s). Although the claims are interpreted in light of the specification, limitations from the specification are not read into the claims. See *In re Van Geuns*, 988 F.2d 1181, 26 USPQ2d 1057 (Fed. Cir. 1993).

13. Applicant argues on the second paragraph of page 9 that Argesen '125 is describing what happens in the running program, not where to store stack-map data, which the running program has no knowledge of, and that in the present invention, stack space is being reserved so that the computed stack maps may be placed when the program analysis is finished. In response, Examiner respectfully disagrees. Agesen '125 discloses using a "stack map" to map controls to a "program counter" (col 7, lines 1-5). There is an "active frame pointer" that is located on the stack and is used in part of the mapping. The pointer can be addressed at any of the locations (col 8, lines 53-60). It is inherent that the counter counts with integer iterations just like every other program counter can. For example, iterator "i" will keep iterating in a loop until the "if statement" condition is satisfied and if not, the iterator can be increased or decreased by 1. Specifically, if i is initially set to 0, and the "if statement" condition is "i=4," the iterator will complete the loop without satisfying the condition, update the iterator to "i=1," and keep looping until the iterator "i" completes the loop at "i=4."

14. Applicant argues on the first paragraph of page 10 that the cited passage from O'Connor does not indicate anything about the stack map being placed in the heap within the compiled method, and in general, O'Connor discusses ordinary stores to the heap but not the storage of the stack map. In response, the Examiner respectfully disagrees. O'Connor teaches dynamic storage allocation of heaps that can be done for garbage collection ("dynamic allocation", "dynamically allocated storage", "garbage collection is particularly attractive", "heap-allocated storage", col. 1, lines 10-51). It is shown that stacks are also used and mapped in the garbage collection

("stores to the heap", "stack represented at stack cache 155 are effectively masked by the 32-bit garbage collection page mask", "virtual memory page", col. 9, lines 42-67).

Conclusion

THIS ACTION IS MADE FINAL. Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Kenneth Tang whose telephone number is (703) 305-5334. The examiner can normally be reached on 8:30am-6:00pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Alvin Oberley can be reached on (703)305-9716. The fax phone numbers for the organization where this application or proceeding is assigned are none for regular communications and none for After Final communications.

Any inquiry of a general nature or relating to the status of this application or proceeding should be directed to the receptionist whose telephone number is none.

kt
February 11, 2003

MAJID BANANKHAH
PRIMARY EXAMINER